UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/659,762	09/10/2003	Bruce W. Warila	SIMQ-110-US	4844

45464 7590 07/03/2007

JACOBS & KIM LLP
1050 WINTER STREET
SUITE 1000, #1082
WALTHAM, MA 02451-1401

EXAMINER	
TECKLU, ISAAC TUKU	
ART UNIT	**PAPER NUMBER**
2192	
MAIL DATE	**DELIVERY MODE**
07/03/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/659,762	WARILA ET AL.	
	Examiner	Art Unit	
	Isaac T. Tecklu	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 September 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-94* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-94* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *05/05/04*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed on 09/10/2003.

2. Claims 1-94 have been examined.

Oath/Declaration

3. The oath or declaration is defective. A new oath or declaration in compliance with 37 CFR 1.67(a) identifying this application by application number and filing date is required. See MPEP §§ 602.01 and 602.02.

The oath or declaration is defective because:

It does not identify the mailing address of each inventor. A mailing address is an address at which an inventor customarily receives his or her mail and may be either a home or business address. The mailing address should include the ZIP Code designation. The mailing address may be provided in an application data sheet or a supplemental oath or declaration. See 37 CFR 1.63(c) and 37 CFR 1.76.

It was not executed in accordance with either 37 CFR 1.66 or 1.68.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-76 and 79-94 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wong et al. (US 2003/0067485 A1) in view of Dietz et al. (US 6,754,676 B2), further in view of Lewallent (US 6,941,520 B1).

Per claim 1, Wong discloses a method for enabling the creation and management of platform-independent applications, the method comprising:

generating a platform-independent data superstructure defining the appearance of an application independent of characteristics of a digital processing device on which the application is to be instantiated (paragraph [0110] "... consistent appearance ..." and e.g. FIG. 11, 250, 252 "DEVICE PLATFORM GENERATER IR TREE" and related text).

instantiating the application in the device in accordance with the superstructure (e.g. FIG. 11, 254 "INSTANTIATE IR TREE" and related text),

updating, in response to generated application events, information in a segment of the superstructure associated with the application events, the application events including events generated by the application instantiated in the device and representative of an application state, and updating, in accordance with the superstructure segment update, the application state in the device, wherein (e.g. FIG. 15, 506 "UPDATE PLATFORM INDEP. PRES. MODEL WITH RUNNING STATE" and related text):

application events are expressed to the superstructure via a pathway including a device-native operating system (OS) and a superstructure-dedicated OS acting as an intermediary between the device-native OS and the superstructure (e.g. FIG. 10, 204 and 202 and related text), whereby:

a defined portion of the application can be addressed and updated in response to application events without necessitating update of the entire application, and the appearance and behavior of the application can be propagated with consistency across heterogeneous device types, to enable cross-device interoperability, replicability, and compatibility of applications and data with a consistency of user experience (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with

different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

generating a platform-independent data superstructure defining the appearance of an application independent of characteristics of a digital processing device on which the application is to be instantiated (paragraph [0110] "... consistent appearance ..." and e.g. FIG. 11, 250, 252 "DEVICE PLATFORM GENERATER IR TREE" and related text).

Wong does not explicitly disclose the superstructure is an XML information structure, application appearance and behavior are encapsulated within the superstructure. However, Dietz discloses XML and tree data structure are used for providing selective view in user interface. Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Dietz to use XML for platform independent as XML assures that the data is easily portable to many different platforms and also assures that data structure is able to be parsed and entered into any database quickly as once suggested by Dietz (in col. 5: 20-35).

Wong does not explicitly disclose data superstructure defining the behavior of an application independent of characteristics of a digital processing device on which the application is to be instantiated. However Lewallent discloses providing distinctive platform-independent appearance and standard behavior (col. 1:51-65). Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Lewallent to allow application to access objects outside of the area of execution as once suggested by Lewallent (col. 1; 51-65) and allowing application program to execute across different operating system platforms (col. 3:20-30).

Per claim 2, Wong discloses a method for enabling the creation and management of platform-independent applications, the method comprising:

generating a platform-independent data superstructure defining the appearance of an application independent of characteristics of a digital processing device on which the application is to be instantiated (paragraph [0110] "... consistent appearance ..." and e.g. FIG. 11, 250, 252 "DEVICE PLATFORM GENERATER IR TREE" and related text).

instantiating the application in the device in accordance with the superstructure (e.g. FIG. 11, 254 "INSTANTIATE IR TREE" and related text),

updating, in response to generated application events, information in a segment of the superstructure associated with the application events, the application events including events generated by the application instantiated in the device and representative of an application state, and updating, in accordance with the superstructure segment update, the application state in the device, wherein (e.g. FIG. 15, 506 "UPDATE PLATFORM INDEP. PRES. MODEL WITH RUNNING STATE" and related text):

application events are expressed to the superstructure via a pathway including a device-native operating system (OS) and a superstructure-dedicated OS acting as an intermediary between the device-native OS and the superstructure (e.g. FIG. 10, 204 and 202 and related text), whereby:

a defined portion of the application can be addressed and updated in response to application events without necessitating update of the entire application, and the appearance and behavior of the application can be propagated with consistency across heterogeneous device types, to enable cross-device interoperability, replicability, and compatibility of applications and data with a consistency of user experience (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Wong does not explicitly disclose the superstructure is a hierarchical information structure, application appearance and behavior are encapsulated within the superstructure. However, Dietz discloses XML and tree data structure are used for providing selective view in user interface. Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Dietz to use XML for platform independent as XML assures that the data is easily portable to many different platforms and also assures that data structure is able to be parsed and entered into any database quickly as once suggested by Dietz (in col. 5: 20-35).

Wong does not explicitly disclose data superstructure defining the behavior of an application independent of characteristics of a digital processing device on which the application is to be instantiated. However Lewallent discloses providing distinctive platform-independent appearance and standard behavior (col. 1:51-65). Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Lewallent to allow application to access objects outside of the area of execution as once suggested by Lewallent (col. 1;51-65) and allowing application program to execute across different operating system platforms (col. 3:20-30).

Per claim 3, Wong discloses the method of claims 1 or 2 further comprising:

generating a message containing a data object of a defined type operable to instantiate the application in a device (paragraph [0079] "... creating messages ..."),

transmitting the message to a device operable to instantiate the application in accordance with the data object, receiving the message at the device (e.g. FIG. 11, 258 and related text), and

instantiating the application in the device in accordance with the data object in the received message (e.g. FIG. 11, 260 and related text).

Per claim 4, Wong discloses the method of claim 3 wherein the instantiating of the superstructure inside the target device occurs substantially when the application is invoked (paragraph [0175] "... invoke an action ...").

Per claim 5, Wong discloses the method of claim 3 wherein the instantiating of the superstructure inside the target device occurs at an application provisioning time prior to application run-time (paragraph [0215] "... prior to a migration the running states ...").

Per claim 6, Wong discloses the method of claim 3 further wherein: a provisioning application on a first device locates within its operating environment a first superstructure for a new application superstructure to be expressed to a second device (e.g. FIG. 8, 118 and related text);

the provisioning application generates a defined data object to be used to express the new application superstructure to the second device (e.g. FIG. 8, 125, 122 and related text); the data object is transmitted to the second device via a message; and the second device creates a new application superstructure from the data object in the message (e.g. FIG. 8, 124 and related text).

Per claim 7, Wong discloses the method of claim 3 further wherein: a provisioning application on a first device locates within its operating environment a predefined data object that expresses a new application superstructure for a second device; the predefined data object is transmitted to the second device via a message; and the second device creates its own copy of the new application superstructure from the data object in the message (e.g. FIG. 8, 124 and related text).

Per claim 8, Wong discloses the method of claim 3 further wherein:

a first device maintains an application capable of accepting input from a user to create an interactive message; the first application translates an operational portion of the message into a new superstructure-based application operable to display the message or cause interactive operations within the message; and the first application initiates the transmission of the superstructure of the new application to a receiving device (e.g. FIG. 15, 502-528 and related text).

Per claim 9, Wong discloses the method of claim 3 further wherein: the transmission of the superstructure includes converting the superstructure into a temporary form that is transmitted, received, and decoded back into an original form on the receiving device (e.g. FIG. 15, 502-516 and related text); and the receiving device maintains an application that receives the superstructure in its temporary form, decodes it, and causes the message-bearing superstructure to operate, thereby rendering the message (e.g. FIG. 15 and related text).

Per claim 10, Wong discloses the method of claims 1 or 3 wherein, for a given state of a selected application, the organization of the superstructure is substantially invariant, regardless of the device, platform or device-native operating system environment in which the associated application is instantiated, so as to maintain a consistent application appearance and behavior

across heterogeneous devices, platforms or device-native operating system environments experience (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 11, Wong discloses the method of claims 1 or 3 wherein the superstructure defines rules of appearance and behavior of the application which are substantially invariant across heterogeneous devices, platforms or device-native operating system environments experience (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 12, Wong discloses the method of claims 1 or 3 wherein substantially identical application source code can be used across heterogeneous devices, platforms or device-native operating system environments (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 13, Wong discloses the method of claim 1 or 3 wherein the application includes a user interface, and wherein the user interface has a substantially identical appearance and behavior across heterogeneous devices, platforms or device-native operating system environments experience (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 14, Wong discloses the method of claim 1 wherein operation of the application is implemented through operations on the superstructure, and wherein the operation comprises: receiving an application event in the device-native OS, receiving data representative of the application event in the superstructure-dedicated OS, applying to the superstructure, in response to the received data (paragraph [0133] '... receives the IR tree from the ..."), a data object,

thereby modifying the superstructure, and operating the application in the device in accordance with the modified superstructure (paragraph [0133] "... and initiates the transformation ...").

Per claim 15, Wong discloses the method of claim 14 further comprising generating a modification data object representative of the modification to be applied to the superstructure, translating the modification data object into a form suitable for processing by the device-native OS, receiving in the device-native OS the translated modification data object, and processing the translated modification data object in the application to update the application (e.g. FIG. 10, 204 and 202 and related text).

Per claim 16, Wong discloses the method of claim 15 further comprising expressing within the superstructure a mechanism for generating the modification data object (paragraph [0133] "... and initiates the transformation ...").

Per claim 17, Wong discloses the method of claim 15 wherein modifying the superstructure includes transmitting a portion of the superstructure to a processor remote from the device, modifying the transmitted portion, and then returning the modified portion or a new set of operations to update the superstructure (e.g. FIG. 10, 204 and 202 and related text).

Per claim 18, Wong discloses the method of claim 15 wherein modifying the superstructure includes using device-native code to implement an interface to modify the superstructure (e.g. FIG. 11, 258 and related text).

Per claim 19, Wong discloses the method of claim 15 wherein the application of changes to the superstructure is implemented by activating program instructions within the superstructure (e.g. FIG. 11, 260 and related text).

Per claim 20, Wong discloses the method of claim 1 or 3 wherein: a copy of the superstructure is stored on an application server operable to communicate with a remote device across a network comprising the application server, the remote device, and a communications

channel there between, and the superstructure can include data objects operable to instantiate applications in the remote device (e.g. FIG. 3 and related text), the method further comprising: providing communication of applications between the application server and the remote device by replicating data objects in the superstructure to the remote device via the communications channel, so as to enable instantiation of new data objects and applications from the server into the remote device (e.g. FIG. 3, 206 and related text).

Per claim 21, Wong discloses a method for enabling the creation and management of platform-independent applications, the method comprising:

generating a platform-independent data superstructure defining the appearance and behavior of an application independent of characteristics of a digital processing device on which the application is to be instantiated (e.g. FIG. 11, 250, 252 "DEVICE PLATFORM GENERATER IR TREE" and related text),

instantiating the application in the device in accordance with the superstructure (e.g. FIG. 11, 254 "INSTANTIATE IR TREE" and related text),

updating, in response to generated application events, information in a segment of the superstructure associated with the application events, the application events including events generated by the application instantiated in the device and representative of an application state, and updating, in accordance with the superstructure segment update, the application state in the device, wherein (e.g. FIG. 15, 506 "UPDATE PLATFORM INDEP. PRES. MODEL WITH RUNNING STATE" and related text):

the superstructure is a hierarchical information structure, application appearance and behavior are encapsulated within the superstructure (e.g. FIG. 11, 254 and related text), and

application events are expressed to the superstructure via a pathway including a device-native operating system (OS) and a superstructure-dedicated OS acting as an intermediary between the device-native OS and the superstructure (e.g. FIG. 10, 204 and 202 and related text), whereby:

a defined portion of the application can be addressed and updated in response to application events without necessitating update of the entire application, and the appearance and behavior of the application can be propagated with consistency across heterogeneous device

types, to enable cross-device interoperability, replicability, and compatibility of applications and data with a consistency of user experience (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ..."), and further wherein:

when an application event is expressed to the superstructure, a superstructure object associated with the application event is transmitted via a communications pathway from the device to a remote server (e.g. FIG. 10 and related text),

the server processes the object and creates a new version of the object, responsive to the application event (e.g. FIG. 10, 206 and related text),

the new version of the object is transmitted from the server to the device to replace the existing version of the superstructure object, thus updating the superstructure, and

the superstructure-dedicated OS causes the device-native OS to update the application state in response to the updated superstructure (e.g. FIG. 10, 204 and 202 and related text).

Wong does not explicitly disclose the superstructure is an XML information structure, application appearance and behavior are encapsulated within the superstructure. However, Dietz discloses XML and tree data structure are used for providing selective view in user interface. Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Dietz to use XML for platform independent as XML assures that the data is easily portable to many different platforms and also assures that data structure is able to be parsed and entered into any database quickly as once suggested by Dietz (in col. 5: 20-35).

Wong does not explicitly disclose data superstructure defining the behavior of an application independent of characteristics of a digital processing device on which the application is to be instantiated. However Lewallent discloses providing distinctive platform-independent appearance and standard behavior (col. 1:51-65). Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Lewallent to allow application to access objects outside of the area of execution as once suggested by Lewallent (col. 1; 51-65) and allowing application program to execute across different operating system

platforms (col. 3:20-30).

Per claim 22, Wong discloses the method of claim 20 wherein the network further comprises a plurality of heterogeneous devices, communications channels and communications providers servicing the communications channels, and wherein the superstructure defines a given application to have an appearance and behavior that can be propagated with consistency across heterogeneous devices, communications channels and communications providers, to enable cross-device interoperability, replicability, and compatibility of applications and data with a consistency of user experience (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 23, Wong discloses the method of claim 22 wherein: the superstructure can be substantially free of device-specific data, modifications to the superstructure can be made in a substantially device-independent manner, and a real-time image of an application running in a first device can be expressed across the network from the first device to a second device to yield a viable instantiation of the application in the second device, regardless of device environment, and wherein the organization of the superstructure and the meaning of objects within it remains substantially constant between instantiations in various device environments (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 24, Wong discloses the method of claim 20 wherein:

the superstructure is capable of completely expressing the running state and functionality of an application operating in a first device, and the application can be substantially identically instantiated into a second device, without loss of state or functionality, by expressing the superstructure into the second device (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different

heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 25, Wong discloses the method of claim 1 further comprising validating the superstructure upon or after modification (e.g. FIG. 21, 654 and related text).

Per claim 26, Wong discloses the method of claim 1 further comprising validating the superstructure after modifying the superstructure, the validation including validation of data updated by processing of an event, so that the modified superstructure cannot express a harmful change to the device-native OS (e.g. FIG. 10, 204 and 202 and related text).

Per claim 27, Wong discloses the method of claim 1 further wherein an application defined by the superstructure can produce external changes only by invoking operations that operate on the superstructure (e.g. FIG. 10, 204 and 202 and related text).

Per claim 28, Wong discloses the method of claim 1 further including providing an interface between an application and a system service, wherein the interface is defined by interaction between the superstructure and the superstructure-dedicated OS (e.g. FIG. 10, 204 and 202 and related text).

Per claim 29, Wong discloses an information processing language adapted to interface with the structure defined in any of claims 1 or, the language being expressible entirely within the superstructure and capable of expressing a set of transformations within the superstructure (e.g. FIG. 10, 204 and 202 and related text), and capable of utilizing and modifying data only within the superstructure, so that: applications utilizing the language cannot affect the state of other applications or operate outside a bounded application container to affect an underlying device platform (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 30, Wong discloses the method of claim 1 wherein the superstructure can contain stylesheets for defining selected application or presentation characteristics (paragraph [0110]).

Per claim 31, Wong discloses the method of claim 30 further comprising configuring stylesheets on a per-device basis (e.g. FIG. 4 and related text).

Per claim 32, Wong discloses the method of claim 30 further comprising configuring stylesheets on a per-group-of-devices basis (paragraph [0110]).

Per claim 33, Wong discloses the method of claim 30 further comprising expressing stylesheets within the superstructure, independent of device-specific limitations (e.g. FIG. 4 and related text).

Per claim 34, Wong discloses the method of claim 30 further comprising selecting a stylesheet at runtime (paragraph [0110]).

Per claim 35, Wong discloses the method of any of claims 1 or 3 wherein an application defined by the superstructure can be transmitted via a peer to peer transaction from a first device in which the application is instantiated, to a second device for instantiation in the second device (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 36, Wong discloses the method of claims 1 or 3, further comprising:

converting at least a portion of the superstructure into a device-portable form, independent of the present state of the application; and reconstructing the original superstructure portion, on the same or different device context, using the device portable form, without loss of state (e.g. FIG. 21 and related text).

Per claim 37, Wong discloses the method of claim 36 wherein the reconstructing includes utilizing a new device-specific stylesheet.

Per claim 38, Wong discloses a method for enabling the creation and management of platform-independent applications, the method comprising:

instantiating the application in the device in accordance with the superstructure (e.g. FIG. 11, 254 "INSTANTIATE IR TREE" and related text),

updating, in response to generated application events, information in a segment of the superstructure associated with the application events, the application events including events generated by the application instantiated in the device and representative of an application state, and updating, in accordance with the superstructure segment update, the application state in the device, wherein (e.g. FIG. 15, 506 "UPDATE PLATFORM INDEP. PRES. MODEL WITH RUNNING STATE" and related text):

application events are expressed to the superstructure via a pathway including a device-native operating system (OS) and a superstructure-dedicated OS acting as an intermediary between the device-native OS and the superstructure (e.g. FIG. 10, 204 and 202 and related text), whereby:

a defined portion of the application can be addressed and updated in response to application events without necessitating update of the entire application, and the appearance and behavior of the application can be propagated with consistency across heterogeneous device types, to enable cross-device interoperability, replicability, and compatibility of applications and data with a consistency of user experience (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

generating a platform-independent data superstructure defining the appearance of an application independent of characteristics of a digital processing device on which the application is to be instantiated (paragraph [0110] "... consistent appearance ..." and e.g. FIG. 11, 250, 252 "DEVICE PLATFORM GENERATER IR TREE" and related text).

Wong does not explicitly disclose data superstructure defining the behavior of an application independent of characteristics of a digital processing device on which the application is to be instantiated. However Lewallent discloses providing distinctive platform-independent appearance and standard behavior (col. 1:51-65). Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Lewallent to allow application to access objects outside of the area of execution as once suggested by Lewallent (col. 1; 51-65) and allowing application program to execute across different operating system platforms (col. 3:20-30).

Wong does not explicitly disclose the superstructure is an XML information structure, application appearance and behavior are encapsulated within the superstructure. However, Dietz discloses XML and tree data structure are used for providing selective view in user interface. Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Dietz to use XML for platform independent as XML assures that the data is easily portable to many different platforms and also assures that data structure is able to be parsed and entered into any database quickly as once suggested by Dietz (in col. 5: 20-35).

Per claim 39, Wong discloses the method of claim 36 further comprising:

using the device-portable form as an intermediate or permanent storage format for recording data within the superstructure (e.g. FIG. 11, 254 "INSTANTIATE IR TREE" and related text).

Per claim 40, Wong discloses the method of any of claims 1 or 3 wherein the superstructure is organized into objects and classes (paragraph [0160]).

Per claim 41, Wong discloses a method for enabling the creation and management of platform-independent applications, the method comprising:

instantiating the application in the device in accordance with the superstructure (e.g. FIG. 11, 254 "INSTANTIATE IR TREE" and related text),

updating, in response to generated application events, information in a segment of the superstructure associated with the application events, the application events including events generated by the application instantiated in the device and representative of an application state, and updating, in accordance with the superstructure segment update, the application state in the device, wherein (e.g. FIG. 15, 506 "UPDATE PLATFORM INDEP. PRES. MODEL WITH RUNNING STATE" and related text):

application events are expressed to the superstructure via a pathway including a device-native operating system (OS) and a superstructure-dedicated OS acting as an intermediary between the device-native OS and the superstructure (e.g. FIG. 10, 204 and 202 and related text), whereby:

a defined portion of the application can be addressed and updated in response to application events without necessitating update of the entire application, and the appearance and behavior of the application can be propagated with consistency across heterogeneous device types, to enable cross-device interoperability, replicability, and compatibility of applications and data with a consistency of user experience (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

generating a platform-independent data superstructure defining the appearance of an application independent of characteristics of a digital processing device on which the application is to be instantiated (paragraph [0110] "... consistent appearance ..." and e.g. FIG. 11, 250, 252 "DEVICE PLATFORM GENERATER IR TREE" and related text).

Wong does not explicitly disclose data superstructure defining the behavior of an application independent of characteristics of a digital processing device on which the application is to be instantiated. However Lewallent discloses providing distinctive platform-independent appearance and standard behavior (col. 1:51-65). Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Lewallent to allow application to access objects outside of the area of execution as once suggested by Lewallent

(col. 1; 51-65) and allowing application program to execute across different operating system platforms (col. 3:20-30).

Wong does not explicitly disclose the superstructure is an XML information structure, application appearance and behavior are encapsulated within the superstructure. However, Dietz discloses XML and tree data structure are used for providing selective view in user interface. Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Dietz to use XML for platform independent as XML assures that the data is easily portable to many different platforms and also assures that data structure is able to be parsed and entered into any database quickly as once suggested by Dietz (in col. 5: 20-35).

Per claim 42, Wong discloses the method of claim 3 wherein a first device can transmit to a second device a message containing an application event item, and thereby cause the second device to place the application event item into a processing queue of the second device (e.g. FIG. 21, 666 and related text).

Per claim 43, Wong discloses the method of claim 20 wherein application logic can be distributed across the network by obtaining a portion of the logic from the remote device and transmitting it in a hierarchical form to the server without the necessity of adapting code therefor (e.g. FIG. 11, 254 "INSTANTIATE IR TREE" and related text).

Per claim 44, Wong discloses the method of claim 20 further comprising providing updates to an application's state from the server to a remote device, by defining a minimal change set to the application's state and transferring it across the network from the server to the remote device, without the necessity of adapting code therefor (e.g. FIG. 13 and related text).

Per claim 45, Wong discloses the method of claims 1, 2 or 3 further comprising incorporating media assets into the superstructure, for reference by running applications (e.g. FIG. 13, 304 and related text).

Per claim 46, Wong discloses the method of claims 1, 2 or 3 further comprising incorporating by reference media assets outside the superstructure, for reference by running applications (e.g. FIG. 13, 302 and related text).

Per claim 47, Wong discloses a method for enabling, in a wireless messaging device operable to communicate with a network serviced by a communications carrier, the creation, modification, and management of platform-independent user interfaces and associated display elements, the method comprising:

generating a platform-independent data superstructure having a state defining a display state of the user interface,

instantiating in the wireless device ~~a messaging~~ an application including an associated user interface, the behavior and state of the application and the associated user interface being defined by the superstructure (e.g. FIG. 11, 254 "INSTANTIATE IR TREE" and related text),,

updating, in response to generated application events, a segment of the superstructure associated with the application events, the application events including associated user interface events (e.g. FIG. 15, 506 "UPDATE PLATFORM INDEP. PRES. MODEL WITH RUNNING STATE" and related text),

updating, in accordance with the superstructure segment update, the application state and user interface state in the wireless device, whereby: a defined portion of the application can be addressed and updated in response to application events without necessitating update of the entire application, and the look and behavior of the application can be propagated with consistency across a network of heterogeneous platforms and communications carrier protocols (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Wong does not explicitly discloses data superstructure defining the behavior of an application independent of characteristics of a digital processing device on which the application is to be instantiated. However Lewallent discloses providing distinctive platform-independent appearance and standard behavior (col. 1:51-65). Therefore it would have been obvious to one

skilled in the art at the time of the invention was made to combine Wong and Lewallent to allow application to access objects outside of the area of execution as once suggested by Lewallent (col. 1;51-65) and allowing application program to execute across different operating system platforms (col. 3:20-30).

Wong does not explicitly disclose the superstructure is a hierarchical information structure, application appearance and behavior are encapsulated within the superstructure. However, Dietz discloses XML and tree data structure are used for providing selective view in user interface. Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Dietz to use XML for platform independent as XML assures that the data is easily portable to many different platforms and also assures that data structure is able to be parsed and entered into any database quickly as once suggested by Dietz (in col. 5: 20-35).

Per claim 48, Wong discloses the method of claim 47 wherein:

the superstructure is an XML information structure, application appearance and behavior are encapsulated within the superstructure, application events are expressed to the superstructure via a pathway including a device-native operating system (OS) and a superstructure-dedicated OS acting as an intermediary between the device-native OS and the superstructure, whereby: a defined portion of the application can be addressed and updated in response to application events without necessitating update of the entire application (see the rejection to claim 1), and

the appearance and behavior of the application can be propagated with consistency across heterogeneous device types, to enable cross-device interoperability, replicability, and compatibility of applications and data with a consistency of user experience (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 49, Wong discloses the method of claim 47 or 48 wherein, for a given state of a selected application, the organization of the superstructure is substantially invariant, regardless

of the device, platform or device-native operating system environment in which the associated application is instantiated, so as to maintain a consistent application appearance and behavior across heterogeneous devices, platforms or device-native operating system environments (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 50, Wong discloses the method of claims 47 or 48 wherein the superstructure defines rules of appearance and behavior of the application which are substantially invariant across heterogeneous devices, platforms or device-native operating system environments (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 51, Wong discloses the method of claims 47 or 48 wherein substantially identical application source code can be used across heterogeneous devices, platforms or device-native operating system environments.

Per claim 52, Wong discloses the method of claims 47 or 48 wherein the application includes a user interface, and wherein the user interface has a substantially identical appearance and behavior across heterogeneous devices, platforms or device-native operating system environments (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 53, Wong discloses the method of claim 48 wherein operation of the application is implemented through operations on the superstructure, and wherein the operation comprises: receiving an application event in the device-native OS (e.g. FIG. 13, 302 and related text),

receiving data representative of the application event in the superstructure-dedicated OS,

applying to the superstructure, in response to the received data, a data object, thereby modifying the superstructure (e.g. FIG. 13, 304 and related text), and operating the application in the device in accordance with the modified superstructure (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 54, Wong discloses the method of claim 53 further comprising generating a modification data object representative of the modification to be applied to the superstructure, translating the modification data object into a form suitable for processing by the device-native OS,receiving in the device-native OS the translated modification data object, and processing the translated modification data object in the application to update the application (e.g. FIG. 21 and related text).

Per claim 55, Wong discloses the method of claim 54 further comprising expressing within the superstructure a mechanism for generating the modification data object (e.g. FIG. 21 and related text).

Per claim 56, Wong discloses the method of claim 54 wherein modifying the superstructure includes transmitting a portion of the superstructure to a processor remote from the device, modifying the transmitted portion, and then returning the modified portion or a new set of operations to update the superstructure (e.g. FIG. 15, 506 "UPDATE PLATFORM INDEP. PRES. MODEL WITH RUNNING STATE" and related text).

Per claim 57, Wong discloses the method of claim 54 wherein modifying the superstructure includes using device-native code to implement an interface to modify the superstructure (e.g. FIG. 21 and related text).

Per claim 58, Wong discloses the method of claim 54 wherein the application of changes to the superstructure is implemented by activating program instructions within the superstructure

(e.g. FIG. 15, 506 "UPDATE PLATFORM INDEP. PRES. MODEL WITH RUNNING STATE" and related text).

Per claim 59, Wong discloses the method of claims 47 or 48 further comprising validating the superstructure upon or after modification (e.g. FIG. 11, 252 and related text).

Per claim 60, Wong discloses the method of claim 48 further comprising validating the superstructure after modifying the superstructure, the validation including validation of data updated by processing of an event, so that the modified superstructure cannot express a harmful change to the device-native OS (e.g. FIG. 13 and related text).

Per claim 61, Wong discloses the method of claim 47 or 48 further wherein an application defined by the superstructure can produce external changes only by invoking operations that operate on the superstructure (paragraph [0175] "... invoke an action ...").

Per claim 62, Wong discloses the method of claim 47 further including providing an interface between an application and a system service, wherein the interface is defined by interaction between the superstructure and the superstructure-dedicated OS (e.g. FIG. 13 and related text).

Per claim 63, Wong discloses the method of claims 47 or 48 wherein the superstructure can contain stylesheets for defining selected application or presentation characteristics (paragraph [0110]).

Per claim 64, Wong discloses the method of claim 63 further comprising configuring stylesheets on a per-device basis (paragraph [0110] FIG. 4 and related text).

Per claim 65, Wong discloses the method of claim 63 further comprising configuring stylesheets on a per-group-of-devices basis (paragraph [0110] and FIG. 4 and related text).

Per claim 66, Wong discloses the method of claim 63 further comprising expressing stylesheets within the superstructure, independent of device-specific limitations (paragraph [0110] and FIG. 4 and related text).

Per claim 67, Wong discloses the method of claim 63 further comprising selecting a stylesheet at runtime (e.g. FIG. 4 and related text).

Per claim 68, Wong discloses the method of claims 47 or 48 further comprising incorporating media assets into the superstructure, for reference by running applications (e.g. FIG. 13 and related text).

Per claim 69, Wong discloses the method of claims 47 or 48 further comprising incorporating by reference, media assets outside the superstructure, for reference by running applications (e.g. FIG. 13 and related text).

Per claim 70, Wong discloses the method of claim 47 wherein the application events include requests to modify the user interface (paragraph [0149] "... obtain user interface ...").

Per claim 71, Wong discloses the method of claim 47 wherein the superstructure includes representations of a library of platform-independent user interface template elements, and the application events include requests for one or more template elements (paragraph [0149] "... user interface related capabilities ...").

Per claim 72, Wong discloses the method of claim 71 wherein application events include requests to add, subtract, replace or otherwise modify elements of the user interface using template elements (paragraph [0149] "... user interface related capabilities ...").

Per claim 73, Wong discloses the method of claim 72 wherein application events include requests to enter user-defined content into the user interface (paragraph [0149] "... user interface related capabilities ...").

Per claim 74, Wong discloses the method of claim 47 wherein user interface events are expressed to the superstructure via a pathway including a device-native operating system and a superstructure-dedicated operating system acting as an intermediary to the superstructure (paragraph [0149] "... user interface related capabilities ...").

Wong does not explicitly disclose wherein the superstructure is an XML data structure, and. However Lewallent discloses providing distinctive platform-independent appearance and standard behavior (col. 1:51-65). Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Lewallent to allow application to access objects outside of the area of execution as once suggested by Lewallent (col. 1;51-65) and allowing application program to execute across different operating system platforms (col. 3:20-30).

Per claim 75, Wong discloses the method of claim 74 further comprising enabling the creation of templates at a remote processor for subsequent representation in the superstructure and instantiation in the wireless device (paragraph [0050] "... mobile device...").

Per claim 76, Wong discloses the method of claim 75 wherein the remote processor is a personal computer (e.g. FIG. 13 and related text).

Per claim 79, Wong discloses the method of claim 47 further comprising configuring the user interface to enable a user to view, generate, send and manage messages (paragraph [0149] "... user interface related capabilities ...").

Per claim 80, Wong discloses the method of claim 79 further comprising configuring the user interface to enable a user to generate messages containing any of text, images, sound, or other media content (paragraph [0131]).

Per claim 81, Wong discloses in a digital processing device running at least one application in accordance with a superstructure-based application environment (SBAE), a method of updating the application, the method comprising:

transmitting, via a wireless communications channel accessible by the SBAE, an application update, the application update including a data object operable to update a segment of the superstructure in the SBAE, receiving the data object (e.g. FIG. 11, 258 and related text), and updating the application in accordance with the application update, and wherein: application appearance and behavior are encapsulated within and defined by the superstructure (e.g. FIG. 15, 506 "UPDATE PLATFORM INDEP. PRES. MODEL WITH RUNNING STATE" and related text), and a defined portion of the application can be addressed and updated in response to application events without necessitating update of the entire application (e.g. FIG. 13 and related text).

Per claim 82, Wong discloses the method of claim 81 further comprising updating an SBAE application across a plurality of devices by broadcasting application updates to the plurality of devices (e.g. FIG. 15, 506 "UPDATE PLATFORM INDEP. PRES. MODEL WITH RUNNING STATE" and related text).

Per claim 83, Wong discloses in a network of digital processing devices operable to communicate over a wireless communications channel, a method for enabling the creation and updating of applications, the method comprising:

generating a platform-independent data superstructure defining the appearance of an application independent of characteristics of a digital processing device on which the application is to be instantiated (paragraph [0110] "... consistent appearance ..." and e.g. FIG. 11, 250, 252 "DEVICE PLATFORM GENERATER IR TREE" and related text).

instantiating the application in the one or more digital processing devices in accordance with the superstructure (e.g. FIG. 11, 254 "INSTANTIATE IR TREE" and related text),

updating the application state in the one or more devices, the updating including:

broadcasting to one or more of the devices in the network at least on application update (e.g. FIG. 10, 204 and 202 and related text),

receiving in the one or more devices the application update e.g. FIG. 10, 204 and 202 and related text), and

updating, in the one or more devices, the running application, in accordance with received application update, and wherein (e.g. FIG. 15, 506 "UPDATE PLATFORM INDEP. PRES. MODEL WITH RUNNING STATE" and related text):

Wong does not explicitly disclose data superstructure defining the behavior of an application independent of characteristics of a digital processing device on which the application is to be instantiated. However Lewallent discloses providing distinctive platform-independent appearance and standard behavior (col. 1:51-65). Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Lewallent to allow application to access objects outside of the area of execution as once suggested by Lewallent (col. 1; 51-65) and allowing application program to execute across different operating system platforms (col. 3:20-30).

Wong does not explicitly disclose the superstructure is an XML information structure, application appearance and behavior are encapsulated within the superstructure. However, Dietz discloses XML and tree data structure are used for providing selective view in user interface. Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Dietz to use XML for platform independent as XML assures that the data is easily portable to many different platforms and also assures that data structure is able to be parsed and entered into any database quickly as once suggested by Dietz (in col. 5: 20-35).

Per claim 84, Wong discloses in a network of digital processing devices operable to communicate over a wireless communications channel, a method for enabling the creation and updating of applications or data, the method comprising:

generating a platform-independent data superstructure defining the appearance of an application independent of characteristics of a digital processing device on which the application

is to be instantiated (paragraph [0110] "... consistent appearance ..." and e.g. FIG. 11, 250, 252 "DEVICE PLATFORM GENERATER IR TREE" and related text).

instantiating the application in the one or more digital processing devices in accordance with the superstructure (e.g. FIG. 11, 254 "INSTANTIATE IR TREE" and related text),

updating application state in the one or more devices, the updating including: broadcasting to one or more of the devices in the network at least one update, receiving in the one or more devices the update (e.g. FIG. 15, 506 "UPDATE PLATFORM INDEP. PRES. MODEL WITH RUNNING STATE" and related text), and

a defined portion of the application can be addressed and updated in response to application events without necessitating update of the entire application (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Wong does not explicitly disclose updating, in the one or more devices, the running application, in accordance with received update, and wherein: application appearance and behavior are encapsulated within the superstructure. However Lewallent discloses providing distinctive platform-independent appearance and standard behavior (col. 1:51-65). Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Lewallent to allow application to access objects outside of the area of execution as once suggested by Lewallent (col. 1; 51-65) and allowing application program to execute across different operating system platforms (col. 3:20-30).

Wong does not explicitly disclose the superstructure is an XML information structure, application appearance and behavior are encapsulated within the superstructure. However, Dietz discloses XML and tree data structure are used for providing selective view in user interface. Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Dietz to use XML for platform independent as XML assures that the data is easily portable to many different platforms and also assures that data structure is able to be parsed and entered into any database quickly as once suggested by Dietz (in col. 5: 20-35).

Per claim 85, Wong discloses the method of any of claims 82, 83 or 84 further comprising:

ensuring that each device is in a consistent, known state at the time of broadcasting and that the update remains whole and complete (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Per claim 86, Wong discloses the method of any of claims 82, 83 or 84 further comprising: broadcasting, in an all-or-nothing manner, only complete segments of application update (e.g. FIG. 10, 204 and 202 and related text).

Per claim 87, Wong discloses in a superstructure-based application environment, a method of enabling the storage and recovery of non-conversational data, the method comprising:

generating a platform-independent data superstructure defining the appearance of an application independent of characteristics of a digital processing device on which the application is to be instantiated (paragraph [0110] "... consistent appearance ..." and e.g. FIG. 11, 250, 252 "DEVICE PLATFORM GENERATER IR TREE" and related text), wherein the superstructure can be serialized in whole or in part at any time, instantiating the application in the device in accordance with the superstructure, updating, in response to generated application events, information in a segment of the superstructure associated with the application events, the application events including events generated by the application instantiated in the device and representative of an application state (e.g. FIG. 11, 254 "INSTANTIATE IR TREE" and related text), and

using an internal representation of the superstructure to store private data relating to requests from the application or the state or data type of a superstructure node, wherein the private data is not serialized when the application is paused, halted or migrated, and is stored in a manner conveniently accessible at application runtime, such that this non-conversational data is coherently recoverable so long as the private data can be re-established upon de-serialization, based on public data that has been maintained in the superstructure (e.g. FIG. 10, 204 and 202 and related text).

Wong does not explicitly disclose data superstructure defining the behavior of an application independent of characteristics of a digital processing device on which the application is to be instantiated. However Lewallent discloses providing distinctive platform-independent appearance and standard behavior (col. 1:51-65). Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Lewallent to allow application to access objects outside of the area of execution as once suggested by Lewallent (col. 1; 51-65) and allowing application program to execute across different operating system platforms (col. 3:20-30).

Wong does not explicitly disclose the superstructure is an XML information structure, application appearance and behavior are encapsulated within the superstructure. However, Dietz discloses XML and tree data structure are used for providing selective view in user interface. Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Dietz to use XML for platform independent as XML assures that the data is easily portable to many different platforms and also assures that data structure is able to be parsed and entered into any database quickly as once suggested by Dietz (in col. 5: 20-35).

Per claim 88, Wong discloses a method for enabling the creation, updating and management of platform-independent applications and the storage and recovery of non-conversational data, the method comprising:

generating a platform-independent data superstructure defining the appearance of an application independent of characteristics of a digital processing device on which the application is to be instantiated (paragraph [0110] "... consistent appearance ..." and e.g. FIG. 11, 250, 252 "DEVICE PLATFORM GENERATER IR TREE" and related text), wherein the superstructure can be serialized in whole or in part at any time, instantiating the application in the device in accordance with the superstructure, updating, in response to generated application events, information in a segment of the superstructure associated with the application events, the application events including events generated by the application instantiated in the device and representative of an application state (e.g. FIG. 11, 254 "INSTANTIATE IR TREE" and related text), and

using an internal representation of the superstructure to store private data relating to requests from the application or the state or data type of a superstructure node, wherein the private data is not serialized when the application is paused, halted or migrated, and is stored in a manner conveniently accessible at application runtime, such that this non-conversational data is coherently recoverable so long as the private data can be re-established upon de-serialization, based on public data that has been maintained in the superstructure (e.g. FIG. 10, 204 and 202 and related text).

Wong does not explicitly disclose data superstructure defining the behavior of an application independent of characteristics of a digital processing device on which the application is to be instantiated. However Lewallent discloses providing distinctive platform-independent appearance and standard behavior (col. 1:51-65). Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Lewallent to allow application to access objects outside of the area of execution as once suggested by Lewallent (col. 1; 51-65) and allowing application program to execute across different operating system platforms (col. 3:20-30).

Wong does not explicitly disclose the superstructure is an XML information structure, application appearance and behavior are encapsulated within the superstructure. However, Dietz discloses XML and tree data structure are used for providing selective view in user interface. Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Dietz to use XML for platform independent as XML assures that the data is easily portable to many different platforms and also assures that data structure is able to be parsed and entered into any database quickly as once suggested by Dietz (in col. 5: 20-35).

Per claim 89, Wong discloses a method for enabling the creation and management of platform-independent applications, the method comprising:

generating a platform-independent data superstructure defining the appearance of an application independent of characteristics of a digital processing device on which the application is to be instantiated (paragraph [0110] "... consistent appearance ..." and e.g. FIG. 11, 250, 252 "DEVICE PLATFORM GENERATER IR TREE" and related text).

instantiating the application in the device in accordance with the superstructure (e.g. FIG. 11, 254 "INSTANTIATE IR TREE" and related text),

updating, in response to generated application events, information in a segment of the superstructure associated with the application events, the application events including events generated by the application instantiated in the device and representative of an application state, and updating, in accordance with the superstructure segment update, the application state in the device, wherein (e.g. FIG. 15, 506 "UPDATE PLATFORM INDEP. PRES. MODEL WITH RUNNING STATE" and related text):

application events are expressed to the superstructure via a pathway including a device-native operating system (OS) and a superstructure-dedicated OS acting as an intermediary between the device-native OS and the superstructure (e.g. FIG. 10, 204 and 202 and related text), whereby:

a defined portion of the application can be addressed and updated in response to application events without necessitating update of the entire application, and the appearance and behavior of the application can be propagated with consistency across heterogeneous device types, to enable cross-device interoperability, replicability, and compatibility of applications and data with a consistency of user experience (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

generating a platform-independent data superstructure defining the appearance of an application independent of characteristics of a digital processing device on which the application is to be instantiated (paragraph [0110] "... consistent appearance ..." and e.g. FIG. 11, 250, 252 "DEVICE PLATFORM GENERATER IR TREE" and related text).

Wong does not explicitly disclose the superstructure is an XML information structure, application appearance and behavior are encapsulated within the superstructure. However, Dietz discloses XML and tree data structure are used for providing selective view in user interface. Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Dietz to use XML for platform independent as XML assures that the

data is easily portable to many different platforms and also assures that data structure is able to be parsed and entered into any database quickly as once suggested by Dietz (in col. 5: 20-35).

Wong does not explicitly disclose data superstructure defining the behavior of an application independent of characteristics of a digital processing device on which the application is to be instantiated. However Lewallent discloses providing distinctive platform-independent appearance and standard behavior (col. 1:51-65). Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Lewallent to allow application to access objects outside of the area of execution as once suggested by Lewallent (col. 1; 51-65) and allowing application program to execute across different operating system platforms (col. 3:20-30).

Per claim 90, Wong discloses a method for enabling the creation and management of platform-independent applications, the method comprising:

generating a platform-independent data superstructure defining the appearance of an application independent of characteristics of a digital processing device on which the application is to be instantiated (paragraph [0110] "... consistent appearance ..." and e.g. FIG. 11, 250, 252 "DEVICE PLATFORM GENERATER IR TREE" and related text).

instantiating the application in the device in accordance with the superstructure (e.g. FIG. 11, 254 "INSTANTIATE IR TREE" and related text),

updating, in response to generated application events, information in a segment of the superstructure associated with the application events, the application events including events generated by the application instantiated in the device and representative of an application state, and updating, in accordance with the superstructure segment update, the application state in the device, wherein (e.g. FIG. 15, 506 "UPDATE PLATFORM INDEP. PRES. MODEL WITH RUNNING STATE" and related text):

application events are expressed to the superstructure via a pathway including a device-native operating system (OS) and a superstructure-dedicated OS acting as an intermediary between the device-native OS and the superstructure (e.g. FIG. 10, 204 and 202 and related text), whereby:

a defined portion of the application can be addressed and updated in response to application events without necessitating update of the entire application, and the appearance and behavior of the application can be propagated with consistency across heterogeneous device types, to enable cross-device interoperability, replicability, and compatibility of applications and data with a consistency of user experience (paragraph [0110] "... consistent appearance ..." paragraph [0093] "... heterogynous device platform ..." paragraph [0066] "... compatible with different heterogeneous device platform ..." paragraph [0066] "... device platform operable with ...").

Wong does not explicitly disclose data superstructure defining the behavior of an application independent of characteristics of a digital processing device on which the application is to be instantiated. However Lewallent discloses providing distinctive platform-independent appearance and standard behavior (col. 1:51-65). Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Lewallent to allow application to access objects outside of the area of execution as once suggested by Lewallent (col. 1;51-65) and allowing application program to execute across different operating system platforms (col. 3:20-30).

Wong does not explicitly disclose the superstructure is a hierarchical information structure, application appearance and behavior are encapsulated within the superstructure. However, Dietz discloses XML and tree data structure are used for providing selective view in user interface. Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Dietz to use XML for platform independent as XML assures that the data is easily portable to many different platforms and also assures that data structure is able to be parsed and entered into any database quickly as once suggested by Dietz (in col. 5: 20-35).

Per claim 91, this is the system version of the claimed method discussed above (Claim 1), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 92, this is the system version of the claimed method discussed above (Claim 2), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 93, this is the system version of the claimed method discussed above (Claim 1), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

Per claim 94, this is the system version of the claimed method discussed above (Claim 2), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also obvious.

6. Claims 77 and 78 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wong et al. (US 2003/0067485 A1) in view of Lewallent (US 6,941,520 B1), in view of Dietz et al. (US 6,754,676 B2) further in view of Snyder (US 6,707,475 B1).

Wong does not explicitly disclose configuring the user interface to respond to controls adapted to be actuated by a user's thumbs and configuring the user interface to provide visual, sonic, tactile or other human-perceptible indications in response to commands entered by a user, or other application events.. However, Snyder teaches user interface includes a cursor control device having a touch-pad device with thumb actuation switch located on its side. When employing the device, the user resets a hand on a built-in palm rest to stabilize the hand, positions the fingertip for pointing, and positions the thumb for clicking (col. 5: 25-35). Therefore it would have been obvious to one skilled in the art at the time of the invention was made to combine Wong and Snyder to select data captured by the cursor for selecting and displaying navigational information as once suggested by Snyder (col.2: 15-30).

Conclusion

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Isaac T. Tecklu whose telephone number is (571) 272-7957. The examiner can normally be reached on M-TH 9:300A - 8:00P.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Isaac Tecklu
Art Unit 2192

TUAN DAM
SUPERVISORY PATENT EXAMINER